Comparison of Background Subtraction Techniques Under Sudden Illumination Changes

C.J.F. Reyneke, P.E. Robinson and A.L. Nel
Department of Electrical and Electronic Engineering Science
Faculty of Engineering and the Built Environment
University of Johannesburg
email: corius.reyneke@gmail.com; {philipr, andren}@uj.ac.za

***Abstract*—This paper investigates three background modelling techniques that have potential to be robust against sudden and gradual illumination changes for a single, stationary camera. The first makes use of a modified local binary pattern that considers both spatial texture and colour information. The second uses a combination of a frame-based Gaussianity Test and a pixel-based Shading Model to handle sudden illumination changes. The third solution is an extension of a popular kernel density estimation (KDE) technique from the temporal to spatio-temporal domain using 9-dimensional data points instead of pixel intensity values and a discrete hyperspherical kernel instead of a Gaussian kernel.**

A number of experiments were performed to provide a comparison of these techniques in regard to classfication accuracy.

***Index Terms*—background subtraction, sudden illumination changes.**

I. Introduction

Background subtraction techniques have traditionally been applied to object detection in computer vision systems and have since become a fundamental component for many applications ranging from human pose estimation to video surveillance. The goal is to remove the background in a scene so that only the interesting objects remain for further analysis or tracking. Techniques such as these are especially useful when they can identify object regions without prior information and when they can perform in real-time.

Real-life scenes often contain dynamic backgrounds such as swaying trees, rippling water, illumination changes and noise. While a number of techniques are effective at handling these, sudden illumination changes such as a light source switching on/off or curtains opening/closing continue to be a challenging problem for background subtraction [1]. In recent years a number of new segmentation techniques have been developed that are robust to sudden illumination changes but only for certain scenes. Our aim is to eventually identify the best-performing solution, improve upon it, and implement it on a GPU for real-time application.

II. Related Work

A number of texture-based methods have developed to solve the problem of sudden illumination changes. Heikkila [2], Xie *et al.* [3] and Pilet *et al.* [4] make use of robust texture features [5]. Heikkila makes use of local binary pattern histograms as background statistics. Xie *et al.* assumes that pixel order values in local neighbourhoods are preserved in the presence of sudden illumination changes. They provide an output image by classifying each pixel by its probability of order consistency [3]. Pilet *et al.* make use of texture and colour ratios to model the background and segment the foreground using an expectation-maximization framework [4]. Texture-based features work well, but only in scenes with sufficient texture; untextured objects prove to be a difficulty.

Another way of dealing with sudden illumination changes is to maintain a representative set of background models [5]. These record the appearance of the background under different lighting conditions and alternate between these models depending on observation. The techniques that make use of this approach mostly differ in their method of deciding which model should be used for the current observation. Toyama *et al.* [6], implement the Wallflower system which chooses the model as the one that produces the lowest number of foreground pixels. This proves to be an unreliable criterion for real-world scenes. Stenger *et al.* [7] make use of hidden Markov models but in most cases, sharp changes occur without any discernible pattern. Also, Stenger *et al.* and Toyama *et al.* require off-line training procedures and consequently cannot incorporate new real-world scenes into their models during run-time [8]. Sun [9] implements a hierarchical Gaussian Mixture Model (GMM) in a top-down pyramid structure. At each scale-level a mean pixel intensity is extracted and is matched to the best model of its upper-level GMM. While mean pixel intensity is useful for the detection of illumination changes, it is also sensitive to changes caused by the foreground. Additionally, the Hierarchical GMM does not exploit any spatial relationships among pixels which can output incoherent segmentation [5]. Dong *et al.* [10] employ principle component analysis (PCA) to build a number of subspaces where each represent a single background appearance. The foreground is segmented by selecting the subspaces which produces minimum reconstruction error. However, their work does not discuss how the system reacts to repetitive background movements.

More recently, Zhou *et al.* [11], Ng *et al.* [1] and Vemulapalli [12] have developed techniques that have potential to handle, and even be robust to, sudden illumination changes. These will be discussed in more detail in section III.

III. Proposed Solutions

A. Background Modeling using Spatial-Colour Binary Patterns (SCBP)

This approach makes used of a novel feature extraction operator, the Spatial-Colour Binary Pattern (SCBP), which takes spatial texture and colour information into consideration [11]. It is an extension of a local binary pattern which is adapted to be centre-symmetrical and to consider only two colour channels for the sake of computational efficiency. For the sake of simplicity all processes relating to this solution apply to a single pixel and are performed on all the pixels in an image.

$$\begin{aligned}\mathrm{SCBP}_{2N,R}(x_c,y_c) &= CSLBP_{2N,R}(x_c,y_c)\\ &+2^{N+1}f(R_c,G_c|\gamma)+2^{N+2}f(G_c,B_c|\gamma)\end{aligned} \quad (1)$$

$$f(a,b|\gamma)=\begin{cases}1, & a>\gamma b\\ 0, & \text{otherwise}\end{cases} \quad (2)$$

Where R_c, G_c and B_c are the three colour channels of the centre pixel (x_c, y_c) and $\gamma > 1$ is a noise suppression factor. The Centre-Symmetrical Local Binary Pattern (CSLBP) is defined as:

$$\mathrm{CSLBP}_{2N,R}(x_c,y_c)=\sum_{i=0}^{N-1}2^i s(g_i-g_{i+N}) \quad (3)$$

$$s(x)=\begin{cases}1, & x>=0\\ 0, & x<0\end{cases} \quad (4)$$

Where g_i is the grey value of the neighbouring pixel at index i and N is the number of neighbours to be compared. The neighbours are evenly distributed on a circle around the centre pixel with radius R. If a neighbour value does not fall exactly on a pixel it is estimated using bilinear interpolation.

An SCBP histogram is computed over a circular region of radius R_{region} around the pixel. Using this as a feature vector a model consisting of K SCBP histograms is built, each with their own weight, such that $w_0+w_1+w_K=1.0$ in decreasing order. At the start these model histograms will be identical but will begin to differ as their respective weights are updated.

An SCBP histogram is calculated for each new frame and then compared to the model histograms using a proximity measure. This measure adds the mutual minimum histogram bins of the current frame and each model histogram that comprise the background model. The proximity measure is defined as follows:

$$\cap(\bar{a},\bar{b})=\sum_{n=0}^{N-1}min(a_n,b_n) \quad (5)$$

Where $\bar{a}$ and $\bar{b}$ are histograms and N is the number of bins in each histogram.

The model is updated selectively depending on the value of the calculated proximity measures. If all the proximity measures are below the threshold, T_p, the model histogram with the lowest weight has its bins replaced with those of the current frame. If at least one proximity measure is above the threshold then only the background histogram that produced the highest proximity measure is updated using the following formula:

$$\bar{m_k}=\alpha_b\bar{h}+(1-\alpha_b)\bar{m_k} \quad (6)$$

Where $\bar{m_k}$ is the model SCBP histogram, $\bar{h_k}$ is the current frame SCBP histogram and α_b is a learning rate such that $\alpha_b \in [0,1]$.

Furthermore, the weights of the model are updated as follows:

$$w_k=\alpha_w M_k+(1-\alpha_w)w_k \quad (7)$$

Where α_w is a learning rate such that $\alpha_w \in [0,1]$ and M_k is 1 for the best-matching histogram and 0 for the rest.

T_p is an adaptive threshold that is maintained (for each pixel). The advantage of this is that static regions become more sensitive while dynamic regions have a higher tolerance. The threshold is updated as follows:

$$T_p(x,y)=\alpha_p(s(x,y)-0.05)+(1-\alpha_p)T_p(x,y) \quad (8)$$

Where α_p is a learning rate such that $\alpha_p \in [0,1]$ and $s(x,y)$ is a similarity measure of the highest value between the current frame's SCBP histogram bins and those of the model histograms.

In order to determine the foreground mask the value for n in the following equation is first determined.

$$w_0+w_1+...+w_n \leq T_w \quad (9)$$

Where the weights have been sorted into descending order. T_w is a fixed threshold and is dependent upon the number of histograms that make up the model. The calculated value for n determines the number of corresponding model histograms which are selected to be part of the background model. The advantage of using this weighted technique is that the persistence of a model histogram is directly related its weight. Persistence needs to be considered because all of the model histograms are not necessarily produced by background processes; the bigger the weight of a model histogram, the higher the probability it has of being a background histogram.

If the proximity measure values for all the background model histograms are smaller than the threshold T_w the pixel is classified as background. If the proximity value for at least one of the background models is greater than T_w then the pixel is classified as foreground.

Furthermore, object contours are refined using a statistical operator to reduce false positives. These are based upon two assumptions. A pixel should only be successfully classified as part of the foreground if its intensity value deviates much from the average pixel intensity of its pixel neighbourhood and its colour composition changes much from that of its pixel neighbourhood. Ergo, a binary mask is constructed as follows and is convolved with the output of the foreground detection module:

$$\Omega_i=\begin{cases}1, & \text{if } [d_i>=\xi\mathrm{std}_i]\&[d_i/\bar{g_i}\geq\varepsilon_1],\\ 1, & \text{if}||(r_i,g_i,b_i)-(\bar{r}_i,\bar{g}_i,\bar{b}_i)||_2\geq\varepsilon_2,\\ 0, & \text{otherwise}\end{cases} \quad (10)$$

Where $d_i = abs(g_i - \bar{g}_i)$ is the absolute deviation of intensity from the average and r, g and b are chromaticity coordinates calculated by $r = R/(R+G+B)$, $g = G/(R+G+B)$ and $b = B/(R+G+B)$. The parameters ξ, ε_1 and ε_2 are tuning parameters.

Finally, the average and standard deviation of the resulting background pixels are updated as follows:

$$\bar{g}_i = \beta g_i + (1-\beta)\bar{g}_i \tag{11}$$

$$std_i = \sqrt{\beta(g_i - \bar{g}_i)^2 + (1-\beta)std_i^2} \tag{12}$$

Where β is a learning rate such that $\beta \in [0, 1]$ The chromaticity coordinates, $\bar{r}_i, \bar{g}_i, \bar{b}_i$, are updated in the same way as was done for $\bar{g}_i$.

B. Background Modeling using a Shading Model and a Gaussianity Test

The method proposed by Ng *et al* implements a hierarchical framework that uses a combination of a pixel-based Shading Model and a block-based Gaussianity Test [1]. This approach is based on the assumption that camera noise is both spatially Gaussian, and temporally uncorrelated. If the difference of two consecutive frames are taken, only Gaussian noise and foreground objects should remain. Under these assumptions, they deduce that background pixels will be Gaussian distributed and foreground pixels will be non-Gaussian distributed. Therefore background pixels can be distinguished from foreground pixels using a Gaussianity test.

The Gaussianity Test statistic is defined as follows:

$$H(J_1, J_2, J_4) = J_4 + 2J_1^4 - 3J_2^2 \tag{13}$$

Where J_k is a moment defined by the following equation:

$$\hat{J}_k(x,y) = \frac{1}{M^2} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \sum_{n=1-\frac{M-1}{2}}^{\frac{M-1}{2}} [D_t(x+m, y+n)]^k \tag{14}$$

The Gaussianity Test statistic is expected to be close to zero when a set of samples is Gaussian distributed. If a set of samples in a block of size MxM has a Gaussianity Test statistic that is greater than a predefined threshold, τ, then the block is considered to contain foreground pixels.

$$\text{block} = \begin{cases} \text{foreground}, & \text{if } H > \tau \\ \text{background}, & \text{otherwise} \end{cases} \tag{15}$$

However, this assumption does not perform well in the presence of sudden illumination changes. A shading model is implemented to handle these.

Ng *et al* extend the Gaussianity test with a shading model proposed by Skifstad [13] in order to make it robust to sudden illumination changes. The shading model is necessary because the previous assumption that background regions are Gaussian distributed does not hold true in the presence of sudden illumination changes.

The shading model assumes that a pixel intensity can be decomposed into an illumination value and a shading coefficient. It is also assumed that if there is no physical change between two frames, such as a moving object, then the ratio of pixel intensities will be constant and independent of the shading coefficients of the frames:

$$R(x,y) = \frac{I_1(x,y)}{I_2(x,y)} = \frac{L_{i,1}}{L_{i,2}} \tag{16}$$

Under this assumption, if no foreground objects exist in a difference frame, the ratio of pixel intensities should remain constant and therefore be Gaussian distributed. Now, by employing the shading model as an input to the Gaussianity test module, the background model can be made robust to sudden illumination changes. The equation used to generate the moments used in the Gaussianity Test statistic is modified to make use of the pixel intensity ratio:

$$\hat{J}_k(x,y) = \frac{1}{M^2} \sum_{m=-\frac{M-1}{2}}^{\frac{M-1}{2}} \sum_{n=1-\frac{M-1}{2}}^{\frac{M-1}{2}} [R_{gt}(x+m, y+n)]^k \tag{17}$$

Where

$$R_{gt}(x,y) = \frac{BM_{t-1}(x,y)}{I_t(x,y)} \tag{18}$$

The foreground mask is obtained using the following equations:

$$D_t(x,y) = |I_t(x,y) - BM_{t-1}(x,y)| \tag{19}$$

$$(x,y) \subset \begin{cases} \text{foreground}, & \text{if } D_t(x,y) > T_a \\ \text{background}, & \text{otherwise} \end{cases} \tag{20}$$

Where $BM_t(x,y)$ is the intensity value of the background model at the coordinates (x,y) and time t, $I_t(x,y)$ is the intensity value of the current pixel at the coordinates (x,y) and time t and T_a is an adaptive threshold. This equation is only employed in the foreground blocks as classified by the Gaussianity test.

T_a is an adaptive threshold which is calculated using an automatic, iterative method first proposed by Ridler [14]. This method is computationally inexpensive but has the disadvantage of assuming that the scene is bimodal. This assumption predicts that there will be two distinct brightness regions in the image represented by two peaks in the grey-level histogram of the input image. These regions correspond to the object and its surroundings and so it is then reasonable to select the threshold as the grey-level half-way between these two peaks.

The histogram of the current frame, $I_t(x,y)$ is segmented into two parts using a threshold, $T_{iterate}$, which is first set to the middle value (127) of the range of intensities. For each iteration, the sample means of the foreground pixel intensities and the sample means of the background pixel intensities are calculated and a new threshold is determined as the average of these two means. The iterations stop once the threshold converges on a value, normally within about 4 iterations. The following formula describes this process:

$$T_{k+1} = \frac{\sum_{b=0}^{T_k} bn(b)}{2\sum_{b=0}^{T_k} n(b)} + \frac{\sum_{b=T_{k+1}}^{N} bn(b)}{2\sum_{b=T_{k+1}}^{N} n(b)} \tag{21}$$

Where T_k is the threshold at the k^{th} iteration, b is the intensity value and $n(b)$ is the number of occurrences of the value b in the image such that $0 \leq b \leq N$.

Once the foreground mask has been segmented, morphological filtering is performed on the foreground mask in order to remove noise. Ng *et al.* perform one closing operation followed by one opening operation.

The values of the background pixels are updated using the following formula:

$$BM_t(x,y) = \begin{cases} BM_{t-1}(x,y), & \text{if } D_t(x,y) \geq T_a \\ I_t(x,y), & \text{if } D_t(x,y) < T_f \\ \alpha I_t(x,y) + \\ (1-\alpha)BM_{t-1}(x,y), & \text{if } T_f \leq D_t(x,y) < T_a \end{cases} \quad (22)$$

Where T_f is fixed and smaller than T_a and α is a learning rate such that $\in [0,1]$

C. Background Modeling using Non-parametric Kernel Density Estimation

The solution proposed by Vemulapalli is an extension of the popular kernel density estimation (KDE) technique first proposed by Elgammal *et al.* [15]. They extend the background model from the temporal to spatio-temporal domain by using 3x3 blocks centred at each pixel as 9-dimensional data points instead of individual pixel intensity values [12]. In order to overcome the obvious increase in computational complexity that this would cause, a hyper-spherical kernel is used instead of the typical Gaussian kernel. Each pass of the background modeling module entails comparing the data points of the current frame, $F_0(x,y)$ with those of the previous frames, $F_{i...N}(x,y)$ selected from a window of size $N = 50$. The Euclidean distance is then employed to compare the data points instead of the typical pixel subtraction as used by Elgammal *et al.* Furthermore, two non-parametric background models, long-term and short-term, in order to exploit their respective advantages at eliminating false positive detections.

So, for each new frame a series of $N-1$ Euclidean distances are calculated by comparing each current pixel's data point to its past data-point values. The higher the value of a Euclidean distance, the higher the probability that the current pixel is part of the foreground. These distances are then thresholded to determine if they lie within the radius of the discrete hyperspherical kernel. This radius is a function of the amount of variation present in the background.

$$M = \sum_{i=1}^{N} \phi\left(\frac{||F_0(x,y) - F_i(x,y)||}{r}\right) \quad (23)$$

Where r is the radius of the hyper-sphere and

$$\phi(u) = \begin{cases} 1, & \text{if } u \leq 1, \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

$||F_0(x,y) - F_i(x,y)||$ is the Euclidean distance between the data points $F_0(x,y)$ and $F_i(x,y)$.

The $N-1$ binary outputs of this module are then summed to produce a type of confidence measure, M of whether the current pixel belongs to the background. This sum is then thresholded using a value, T:

$$\frac{M}{N} \leq T \quad (25)$$

The long-term and short-term models are updated using a blind update and selective update mechanism respectively. The blind update adds a new 9-dimensional data point, $F_i(x,y)$, to the sample set regardless of whether it belongs to the background or foreground while the selective update adds the data-point only if it belongs to the background. When a new data point is added the oldest data point is removed from the sample set. The output of both the long-term and short-term models are used as inputs to the foreground detection module. The output of the module is described by the following table:

Long-term model	Short-term model	Output
$O_l(x,y) = 0$	$O_s(x,y) = 0$	$O_{fd}(x,y) = 0$
$O_l(x,y) = 0$	$O_s(x,y) = 1$	$O_{fd}(x,y) = O'_{fd}(x,y)$
$O_l(x,y) = 1$	$O_s(x,y) = 0$	$O_{fd}(x,y) = 0$
$O_l(x,y) = 1$	$O_s(x,y) = 1$	$O_{fd}(x,y) = 1$

TABLE I: The output of the foreground detection module which combines the output of the short-term and long-term background models.

Where $O_l(x,y) = 1$ is the output of the long-term model, $O_s(x,y) = 1$ is the output of the short-term model and $O_{fd}(x,y) = 1$ is the output of the foreground detection module where:

$$O'_{fd}(x,y) = \begin{cases} 1, & \text{if } \sum_{i=-1}^{1}\sum_{j=-1}^{1} \\ & O_s(x-i,y-j)O_l(x-i,y-j) \\ & \neq 0, \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

If the two models agree on an output, the resultant foreground mask will obviously have the same output. If only the long-term model predicts foreground, the foreground mask will prefer the prediction of the short-term model. In the event of the short-term model predicting foreground and the long-term model predicting a background, a check is performed to see if the two models agree on the output of any of the neighbouring pixels being foreground. If this is the case, the pixel is classified as a foreground.

In the event of a sudden illumination change most of the frame will be classified as foreground and will remain so unitl the long term model adapts to the new lighting conditions. Vemulapalli checks whether more than a certain percentage α of the frame is declared as foreground. If this is the case the short-term model is updated using the blind update mechanism so that it avoids false detections and adapts to the new lighting conditions quickly.

IV. Experimental Methodology

A. Dataset

These techniques will be tested with respect to the accuracy of their outputs. In order to accomplish this three sequences from the publicly available Wallflower dataset [6] are used.

The first sequence is named "Waving Trees" and contains a scene with a typical dynamic background. It has 286 frames where a ground truth is provided for the 247th frame. The second sequence is named "Time of Day" and contains a scene with gradual illumination changes. It has 5889 frames where a ground truth is provided for the 1850th frame. The third sequence is named "Light Switch" and contains a scene with sudden illumination changes. It has 2714 frames where a ground truth is provided for the 1865th frame.

B. Metrics

For the evaluation of the output accuracy we make use of the detection rate (DR), false alarm rate (FAR) and precision (P) statistics. The formulae for these are provided below:

$$DR = \frac{\#true\_positives}{\#true\_positives + \#false\_negatives} \quad (27)$$

$$FAR = \frac{\#false\_positives}{\#false\_positives + \#true\_negatives} \quad (28)$$

$$P = \frac{\#true\_positives}{\#true\_positives + \#false\_positives} \quad (29)$$

Where $\#true\_positives$ is the number of correctly classified foreground pixels, $\#true\_negatives$ is the number of correctly classified background pixels, $\#false\_positives$ is the number of incorrectly classified foreground pixels and $\#true\_negatives$ is the number of incorrectly classified background pixels.

C. Selection of Tuning Parameters

Zhou *et al.* set $R_{region} = 9$, $R = 2$, $N = 4$, $K = 4$, $T_P = 0.65$, $T_B = 0.7$, $\alpha_b = \alpha_w = \beta = 0.01$, $\alpha_p = 0.9$, $\xi = 2.5$ and $\varepsilon_1 = \varepsilon_2 = 0.2$. Zhou *et al* do not specify which similarity measure they used; we investigated two, the L_1 Norm and the Square L_2 Norm. The latter was determined to be best by qualitatively comparing their output. Zhou *et al.* also did not specify how they initialized the weights of the model histograms; we investigated two methods: using a values that decrease linearly and values that decrease exponentially. The latter was determined to be the best by qualitative analysis. Using the exponential curve $w_0 = 0.567$, $w_1 = 0.321$, $w_2 = 0.103$, $w_3 = 0.011$.

Ng *et al.* set $M = 17$ and $\alpha = 0.1$. The value for τ is set empirically for the dataset at hand. For the experiments they perform on the PETS 2006 dataset they set $\tau = 1 \times 10^5$. We set $\tau = 1 \times 10^3$.

Vemulapalli sets $W = 250$, $N = 50$ and $\alpha = 75\%$. However, for the the Waving Trees sequence we set $W = 200$ and $N = 20$ since the 247th frame is used for the ground truth. Vemulapalli does not specify which parameters they used for the hypersphere radius, r, and the threshold, T. We set $r = 1$ and $T = \mu + k\sigma$ where μ is the mean and σ is the standard deviation of the values obtained for M in a frame. k is a positive integer which is set to 6.

V. Experimental Results

A. Waving Trees

From these results shown in *fig. 1* we can see that the Zhou *et al.* provides the best detection rate, moderate precision and worst false alarm rate. Ng *et al.* provides the lowest false alarm rate, but the worst precision and detection rate. Vemulapalli provides the best precision and moderate detection and false alarm rates.

B. Time of Day

From these results shown in *fig. 2* we can see that Zhou has the worst performance; having the worst detection rate, precision and false alarm rate. Ng *et al.* has a superior precision and false alarm rate as well as a moderate detection rate. Vemulapalli provides the best detection rate and values only slightly worse than Ng *et al.* in regard to precision and false alarm rate.

C. Light Switch

From these results shown in *fig. 3* we can see that Zhou *et al.* provides the best detection rate, moderate precision and a moderate false alarm rate. Ng *et al.* has the best precision and false alarm rate, but the worst detection rate. Vemulapalli has a moderate detection rate, but the worst precision and false alarm rate.

The poor performance of the solution proposed by Vemulapalli is largely due to the fact that the sudden illumination check is not triggered by the video sequence. Hence, the blind update mechanism for the short-term model is not employed and the model does not adapt to the new lighting conditions quickly enough.

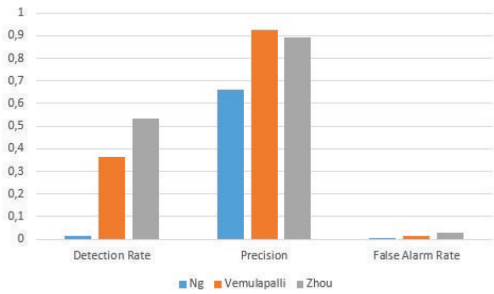

Fig. 1: Results of "Waving Trees" sequence.

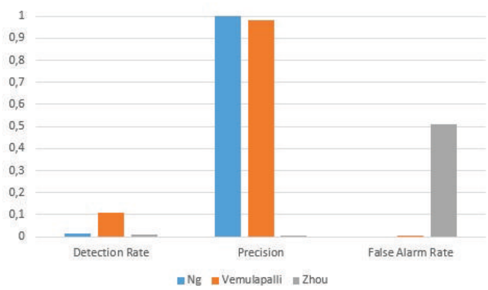

Fig. 2: Results of "Time of Day" sequence.

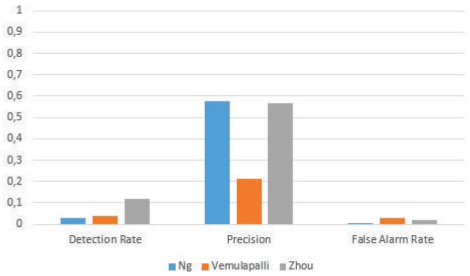

Fig. 3: Results of ”Light Switch” sequence.

Fig. 4: Foreground segmentation masks of proposed solutions. The columns correspond to the ”Waving Trees”, ”Time of Day” and ”Light Switch” sequences respectively. The first row represents the ground truths while the remaining rows correspond to the outputs of the solutions proposed by Zhou *et al.*, Ng *et al.* and Vemulapalli respectively.

VI. Conclusion

This paper investigates three background modelling techniques that are robust against sudden and gradual illumination changes for a single, stationary camera. The first makes use of a modified local binary pattern that considers both spatial texture and colour information. The second uses a combination of a frame-based Gaussianity Test and a pixel-based Shading Model to handle sudden illumination changes. The third solution is an extension of a popular kernel density estimation (KDE) technique from the temporal to spatio-temporal domain using 9-dimensional data points instead of pixel intensity values and a discrete hyperspherical kernel instead of a Gaussian kernel.

A number of experiments were then performed which provide a comparison of these techniques in regard to classification accuracy.

The SCBP histogram feature approach performs well for simple dynamic backgrounds, but not for scenes that contain any type of illumination changes.

The Shading Model and Gaussianity Test approach provides a sparse foreground mask that is very accurate for all three sequences, but has a poor detection rate.

The KDE approach performs well for simple dynamic backgrounds and scenes that contain gradual illumination changes. However, the mechanism employed to handle sudden illumination changes does not work well due to the use of an unreliable criterion for sudden illumination detection.

VII. Future Work

We plan to further investigate the solution proposed by Ng *et al.* and Vemulapalli. Both have potential to be improved through automatic parameter selection and possibly by integrating the strengths of all three the solutions that were investigated.

References

[1] K. K. Ng, S. Srivastava, and E. Delp, “Foreground segmentation with sudden illumination changes using a shading model and a gaussianity test,” in *Image and Signal Processing and Analysis (ISPA), 7th International Symposium on*, September 2011, pp. 236–240.

[2] M. Heikkila and M. Petikainen, “A texture-based method for modelling the background and detecting moving objects,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 657–662, 2006.

[3] B. Xie, V. Ramesh, and T. Boult, “Sudden illumination change detection using order consistency,” *Image and Vision Computing*, vol. 22, no. 2, pp. 117–125, 2004.

[4] J. Pilet, C. Strecha, and P. Fua, “Making background subtraction robust to sudden illumination changes,” in *Computer Vision–ECCV 2008*. Springer, pp. 567–580, 2008.

[5] J. Li and Z. Miao, “Foreground segmentation for dynamic scenes with sudden illumination changes,” *Image Processing, IET*, vol. 6, no. 5, pp. 606–615, July 2012.

[6] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, “Wallflower: principles and practice of background maintenance,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, pp. 255-261, 1999.

[7] B. Stenger, V. Ramesh, N. Paragios, F. Coetzee, and J. M. Buhmann, “Topology free hidden markov models: Application to background modeling,” in *Computer Vision, ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1, pp. 294–301. IEEE, 2001.

[8] X. Zhao, W. He, S. Luo, and L. Zhang, “Mrf-based adaptive approach for foreground segmentation under sudden illumination change,” in *Information, Communications Signal Processing, 2007 6th International Conference on*, pp. 1-4, Dec 2007.

[9] Y. Sun and B. Yuan, “Hierarchical gmm to handle sharp changes in moving object detection,” *Electronics Letters*, vol. 40, no. 13, pp. 801–802, 2004.

[10] Y. Dong, T. Han, and G. N. DeSouza, “Illumination invariant foreground detection using multi-subspace learning,” *International journal of knowledge-based and intelligent engineering systems*, vol. 14, no. 1, pp. 31–41, 2010.

[11] W. Zhou, Y. Liu, W. Zhang, L. Zhuang, and N. Yu, “Dynamic background subtraction using spatial-color binary patterns,” in *Image and Graphics (ICIG), 2011 Sixth International Conference on*, pp. 314-319, Aug 2011.

[12] R. Vemulapalli and R. Aravind, “Spatio-temporal nonparametric background modeling and subtraction,” in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, 1145-1152, Sept 2009.

[13] K. Skifstad and R. Jain, “Illumination independent change detection for real world image sequences,” *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 3, pp. 387–399, June 1989.

[14] T. W. Ridler and S. Calvard, “Picture thresholding using an iterative selection method,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 8, no. 8, pp. 630–632, Aug 1978.

[15] A. M. Elgammal, D. Harwood, and L. S. Davis, “Non-parametric model for background subtraction,” in *Proceedings of the 6th European Conference on Computer Vision-Part II*, ser. ECCV ’00. Springer-Verlag, 2000, pp. 751–767.